参考資料２

SPEC – Power and Performance

User Guide

SPECpower_ssj2008 V1.00

Standard Performance Evaluation Corporation

Table of Contents

1 Introduction

This practical guide will explain how to setup and run the SPECpower_ssj2008 benchmark. In order to submit SPECpower_ssj2008 results, the licensee must adhere to the rules contained in the *Run and Reporting Rules*, which is included in the benchmark kit. For an overview of the benchmark architecture, please see the *SPECpower_ssj2008 Design Document*.

This document is intended for people that wish to run the SPECpower_ssj2008 benchmark in order to accurately measure the power consumption of their server in relation to the server's performance.

1.1 Abstract

SPECpower_ssj2008 is a benchmark suite consisting of three separate software modules:

- Workload (SSJ)
- Power and Temperature Daemon (PTD)
- Control and Collect System (CCS)

These modules work together in real-time to collect server power consumption and performance data by exercising the server under test (SUT) with a predefined workload.

The workload is a Java program designed to exercise the CPU(s), caches, memory, the scalability of shared memory processors, JVM (Java Virtual Machine) implementations, JIT (Just In Time) compilers, garbage collection, threads, and certain aspects of the operating system of the SUT.

The workload architecture is a 3-tier system with emphasis on the middle tier. These tiers are comprised as follows:

1. Random input selection
2. Business logic (fully implemented by SPECpower_ssj2008)
3. Tables of objects, implemented by Java Collections (rather than a separate database)

1.2 General Concepts

1.2.1 Overview

The most basic SPECpower_ssj2008 test bed implementation consists of at least four devices:

- a server under test
- a power analyzer
- a temperature sensor
- a benchmark controller system

Within a given test bed implementation, there are usually several ways to distribute and run the three software modules of the SPECpower_ssj2008 benchmark suite. For simplicity, this section will only consider the basic concepts underlying the simplest possible SPECpower_ssj2008 test bed implementation. Figure 1.2.1-1 illustrates the architecture of a simple SPECpower_ssj2008 benchmark implementation.

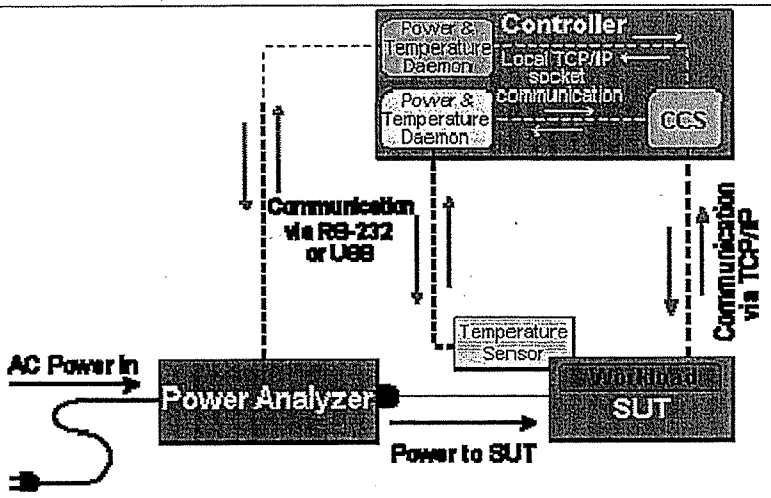


Figure 1.2.1-1

1.2.2 System Under Test (SUT)

The SUT is the system that will be driven by the SSJ (Server Side Java) workload. The SUT's performance and power consumption characteristics will be captured and measured by the benchmark.

1.2.3 Power Analyzer

The power analyzer is the device that is used to measure and capture the power consumption characteristics of the SUT. SPECpower_ssj2008 has a set of minimum criteria that the power analyzer must meet in order to be a candidate for use within a SPECpower_ssj2008 implementation. Please see the Run and Reporting Rules for the minimum power analyzer requirements.

1.2.4 Temperature Sensor

The temperature sensor is used to capture the temperature data of the environment in which the SUT is being tested. There is also a set of minimum requirements that a temperature sensor must meet in order to be used within a valid SPECpower_ssj2008 test bed. Please see Run and Reporting Rules for the minimum temperature sensor requirements.

1.2.5 Controller System

The controller system is a separate system that runs the CCS and Power and Temperature Daemon (PTD) portions of the SPECpower_ssj2008 benchmark suite.

1.3 The SPECpower_ssj2008 Benchmark Suite

1.3.1 Server Side Java (SSJ)

The SPECpower_ssj2008 workload is the portion of the benchmark suite that is installed onto the SUT. It is a Java implementation simulating a standard three-tier client-server architecture designed to exercise several components of the SUT.

The goal of the SPECpower_ssj2008 workload is to drive the SUT throughout the entire spectrum of its inherent power consumption characteristics in a controlled and measurable fashion.

In general, the amount of power required by the SUT (or any computer) is proportional to the amount of work that is done by the SUT.

In SPECpower_ssj2008, the amount of work that is done by the SUT is referred to as throughput. Throughput is the number of transactions that the SUT processes within a given amount of time. Thus, if the SUT's throughput increases, so will the SUT's power consumption. Likewise, if throughput decreases, the SUT's power consumption will also decrease. The general rule is: power consumption varies as a function of throughput.

It follows therefore that in order to control the power consumption characteristics of the SUT, one must control the SUT's throughput characteristics. This is what the SPECpower_ssj2008 workload engine is designed to do. By default, SPECpower_ssj2008 throttles the SUT's throughput from 100% (maximum throughput), down to 0% (no throughput) in steps of 10%. These 10% steps from 100% to 0% throughput are referred to as target loads. (See Figure 1.3.1-1 for a visual reference). In order for the workload engine to accurately execute these target loads during the benchmark run, it must determine the exact throughput values that correspond to each target load. These throughput values are referred to as target throughput.

The technique that the workload engine uses to calculate target throughput for a given target load is as follows: At the beginning of a benchmark run, the workload engine will initiate three calibration intervals (see Figure 1.3.1-1). During these calibration intervals, the workload is allowed to run at maximum throughput ("full throttle"). When the calibration intervals are complete, the workload engine will use the results recorded from the final two calibration intervals to calculate the average throughput that was achieved. This average throughput value is defined as the SUT's maximum level of throughput. (For information concerning why the first calibration interval is omitted in this calculation, see SPECpower_ssj2008 Design Document.)

After the maximum throughput has been determined, the workload engine then calculates the target throughput values that correspond to each target load (100%, 90%, ... 20%, 10%). After this is determined, the benchmark then enters the measurement interval during which the workload iterates through ten target loads. By default, each target load lasts for 240 seconds (not including the ramp up and ramp down periods, which will be covered later).

Finally, after all target load iterations have been completed, the SPECpower_ssj2008 workload will then enter the Active Idle Measurement Interval. In order to more accurately simulate a real-world production environment, it is important to also consider the power consumption characteristics of the SUT while it is in an "idle" state. In SPECpower_ssj2008, the state of "idle" is defined simply as any period of time in which the SUT has reached a quiescent state of 0% throughput. For a more detailed discussion of the Active Idle state and measurement interval, please see the SPECpower_ssj2008 Design Document.

Please reference Figure 1.3.1-1 for a graphical representation of a typical SPECpower_ssj2008 workload iteration.

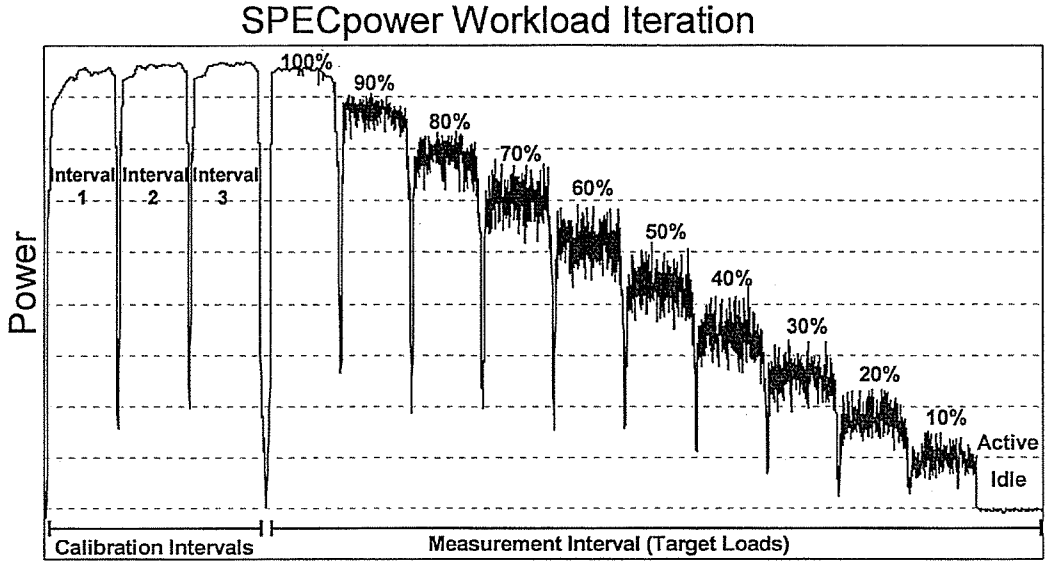


Figure 1.3.1-1

1.3.1.1 SSJ: Architecture

The SSJ workload architecture consists of the following major components:

- JVM Instance
  - Threads (tier 1)
  - Business Logic (tier 2)
  - Warehouses (tier 3)
- JVM Director

The JVM Instance:

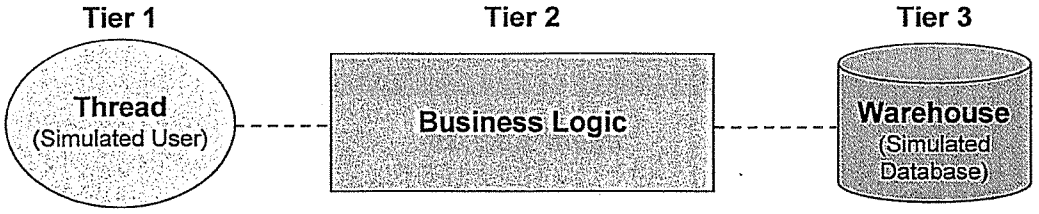


figure 1.3.1.1a

A *JVM instance* is an instance of the SPECpower_ssj2008 workload that runs on the SUT.

A *warehouse* is a unit of stored data. It contains roughly 25MB of data stored in many objects in several collections such as HashMaps and TreeMaps. The warehouse

component of the workload is what actually simulates the third tier of the three-tier client-server architecture, eliminating the need for any type of database back-end.

A *thread* represents an active user posting transaction requests within a warehouse. There is a one-to-one mapping between warehouses and threads. Therefore, the more warehouse instances specified by the user, the more threads will be run concurrently during the benchmark. (How the number of threads/warehouses are specified is covered in section 2.5.1.4)

In a single JVM benchmark run, only one JVM instance is started and run on the SUT. However it is possible and quite common, to run the benchmark in a multi-JVM configuration. This means that more than one workload JVM instance will be started and run concurrently on the SUT. The reason multi-JVM configurations are supported in SPECpower_ssj2008 is because running multiple JVM instances concurrently usually proves to be a very effective method for driving the SUT much closer to its maximum throughput-performance capabilities. For a multi-JVM benchmark run, the final throughput measurement is calculated as the sum of all throughput values achieved by each JVM instance.

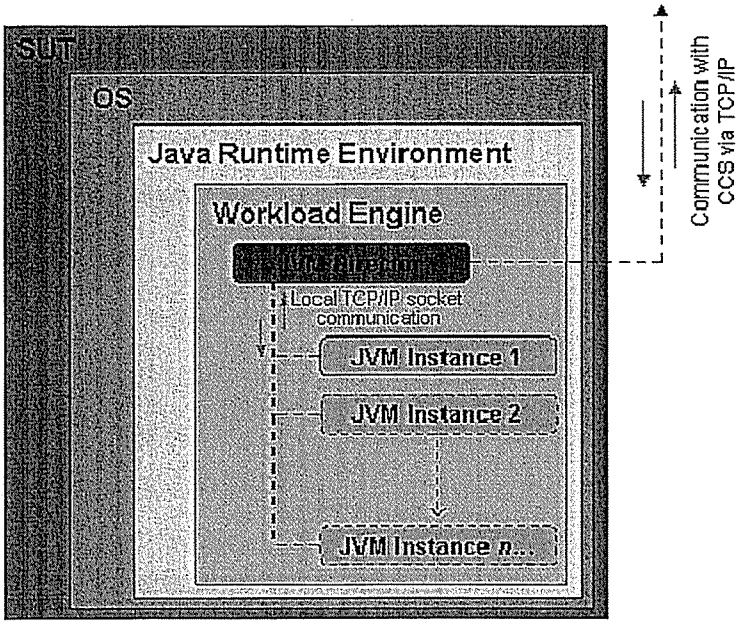


figure 1.3.1.1b

The JVM Director:

The *JVM Director* is a separate and distinct mechanism from the actual workload itself (the three-tiered client-server environment), but runs concurrently with the JVM instance(s) of the workload (see figure 1.3.1.1b). Like the workload, the JVM Director is also a java application, and as such, runs as its own JVM instance. The function of the JVM Director is to:

- *Control the JVM instance(s) of the workload.* The function of the workload is to drive the SUT, however by default the workload instance(s) will not do this autonomously. The workload instance(s) will execute its functions only when

commanded to do so by the JVM Director. The JVM Director and the workload instance(s) communicate via a TCP/IP socket connection(s).

- *Communicate with CCS.* It's the JVM Director's job to receive and distribute commands from CCS. The JVM Director communicates with CCS via a TCP/IP socket connection. CCS will be covered later (in 1.3.3).
- *Collect and report workload data back to CCS.* The JVM Director collects configuration and performance data from the workload instance(s) and then reports this information to CCS when requested to do so by CCS.

The JVM Director can be run locally on the SUT, or it can be run remotely at the user's discretion. Whichever method is employed, the JVM Director and the workload JVM instance(s) will communicate via a TCP/IP socket connection (see figure 1.3.1.1b).

It should be made very clear that in reference to the type of benchmark configuration, the terms single-JVM and multi-JVM refer only to the number of workload JVM instances. Though the JVM Director is an additional JVM instance itself, it is considered part of the benchmark overhead. For a benchmark implementation to qualify as a multi-JVM configuration, there must be two or more workload JVM instances running concurrently. For the remainder of this document, the term *JVM instance* shall be in reference to an SSJ workload instance.

Upon completion of a run, the SPECpower_ssj2008 workload will generate several results reports. These results include throughput data, SUT configuration information, and various other types of workload-specific data. See section 6 for more information.

1.3.2 Power & Temperature Daemon (PTD)

The Power & Temperature Daemon (PTD) is a single executable program that communicates with a power analyzer or a temperature sensor via the server's native RS-232 or USB port. It reports the power consumption or temperature readings to CCS via a TCP/IP socket connection. It supports a variety of RS-232, GPIB and USB interface command sets for a variety of power analyzers and temperature sensors. PTD is the only of the three SPECpower_ssj2008 software modules that is not Java based. Although it can be quite easily setup and run on a server other than the controller server, in the simplest SPECpower_ssj2008 test bed implementation, the PTD will typically reside on the controller server.

1.3.3 Control and Collect System (CCS)

The Control and Collect System is a Java-based application that resides on the controller server. Unlike PTD, CCS must always be run on the controller server. CCS is used to connect to three types of *data sources* via TCP/IP socket communication:

- the SSJ module (via the JVM Director) on the SUT
- an instance of PTD connected to a power analyzer
- an instance of PTD connected to a temperature sensor

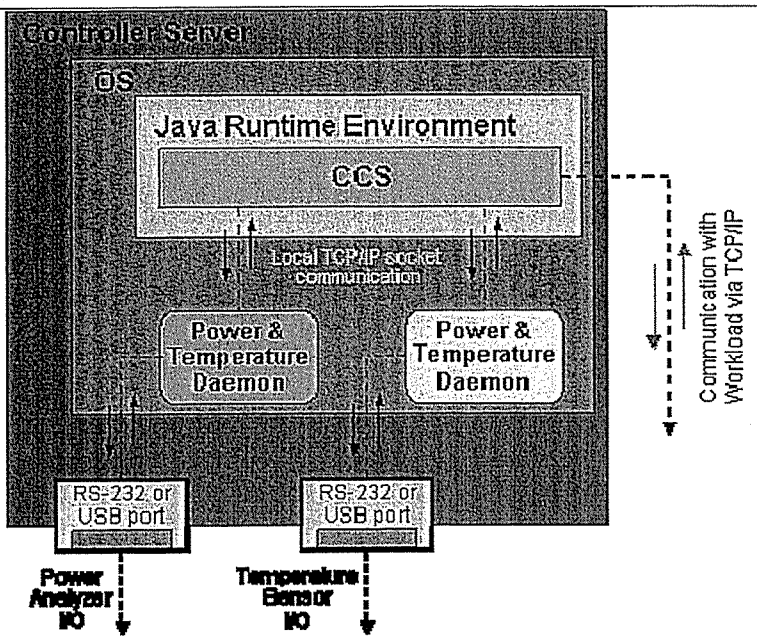


figure 1.3.3

Figure 1.3.3 shows a simple implementation of CCS and two PTD instances all residing on the controller server. Note that as long as reliable TCP/IP communication can be maintained between PTD and CCS, one or both of the PTD instances can be run on another system if the user so desires.

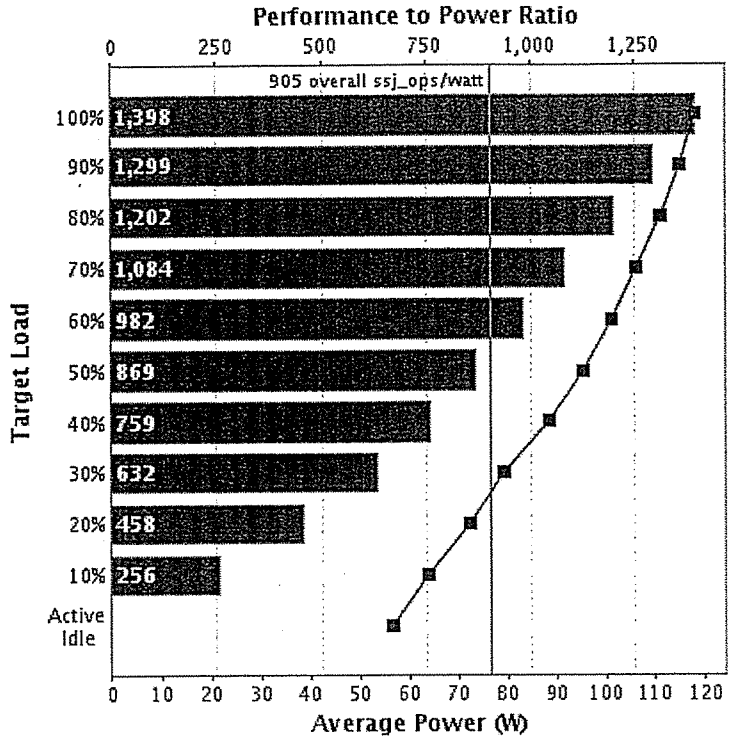
One of CCS's main functions is to initiate the SPECpower_ssj2008 benchmark iteration. After the benchmark run has started, CCS then collects and synchronizes power data, temperature data, and workload data from the three data sources in real time. CCS then outputs this data into several results files (see section 6) for easy viewing by the user.

SPECpower_ssj2008

(Processor, 2.83 GHz)				SPECpower_ssj2008 = 905 overall ssj_ops/watt	
Test Sponsor:	Company Inc.	SPEC License #:	9016	Hardware Availability:	Apr-2008
Tested By:	Company Inc.	Test Location:	Taipei, Taiwan	Software Availability:	Mar-2008
System Source:	Single Supplier	Test Date:	Dec 8, 2008	Publication:	Jan 2, 2009

Benchmark Results Summary

Performance			Power	Performance to Power Ratio
Target Load	Actual Load	ssj_ops	Average Power (W)	
100%	99.6%	165,064	118	1,398
90%	90.1%	149,393	115	1,299
80%	80.7%	133,797	111	1,202
70%	69.5%	115,123	106	1,084
60%	59.9%	99,215	101	982
50%	50.0%	82,814	95.3	869
40%	40.6%	67,254	88.6	759
30%	30.2%	50,013	79.1	632
20%	19.9%	33,046	72.2	458
10%	9.9%	16,358	64.0	256
	Active Idle	0	56.7	0
			∑ssj_ops / ∑power =	905



System Under Test

Hardware	
Hardware Vendor:	Inc.
Model:	Processor, 2.83 GHz
CPU Name:	Processor
CPU Characteristics:	2.83 GHz, 2 x 6 MB L2 Cache, 1333 MHz System Bus
CPU Frequency (MHz):	2833
CPU(s) Enabled:	4 cores, 1 chip, 4 cores/chip
Hardware Threads / Core:	1
CPU(s) Orderable:	1 chip
Primary Cache:	32 KB I + 32 KB D on chip per core
Secondary Cache:	12 MB I+D on chip per chip, 6 MB shared / 2 cores
Tertiary Cache:	None
Other Cache:	None
Memory Amount (GB):	4
# and size of DIMM:	2 x 2048 MB
Memory Details:	DDR2-667 CL5 DIMM; Slots A2 and B2 are populated
Power Supply Quantity and Rating (W):	1 x 220
Power Supply Details:	FSP FSP220-60LE